

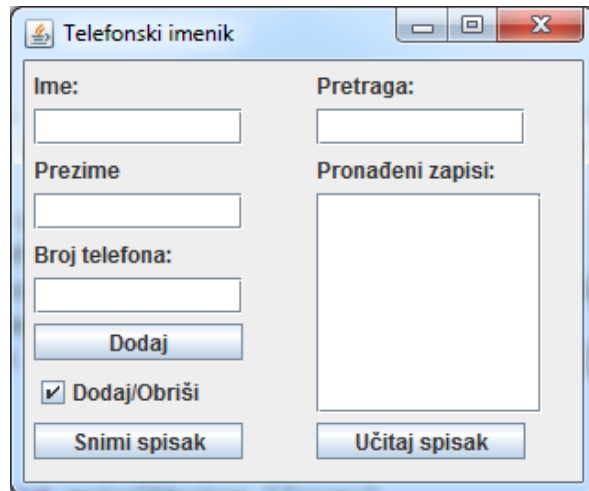
-V čas računskih vježbi-

1. Napisati JAVA program koji će voditi evidenciju o brojevima telefona klijenata. Program treba da omogući korisniku odabir fajla sa podacima o brojevima klijenata, kao i mogućnost brze pretrage po odabranom fajlu. Takođe, aplikacija treba da omogući brisanje i dodavanje kontakata.

```

import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.Scanner;
import javax.swing.DefaultListModel;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextField;
import javax.swing.ListSelectionModel;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
public class Forma extends JFrame
{
    private JLabel [] labele;
    private JTextField [] textPolja;
    private JButton dugme1,dugme2,dugme3;
    private JCheckBox funkcijaDugmeta;
    private JList lista;
    private JPanel p1,p2;
    private DefaultListModel spisak;
    private ArrayList <String> ljudi;
    private String fajl;
    Forma ()
    {
        super("Telefonski imenik");
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setSize(315, 260);
        this.setLayout(new BorderLayout());
        fajl="";
        // Kreiramo dva panela koji će nam služiti za organizaciju komponenti
        p1=new JPanel();
        p1.setPreferredSize(new Dimension(150,260));
        p1.setLayout(new FlowLayout(FlowLayout.LEFT));
        // Prvi panel dodajemo na formu sa lijeve strane
        this.add(p1,BorderLayout.WEST);
        p2=new JPanel();
        p2.setPreferredSize(new Dimension(150,260));
        p2.setLayout(new FlowLayout(FlowLayout.LEFT));
        // Drugi panel dodajemo na formu sa desne strane
        this.add(p2,BorderLayout.EAST);
        // Dodavanje labela na formu
        String naziviLabela[]={"Ime: ", "Prezime", "Broj telefona:", "Pretraga: ", "Pronađeni zapisi: "};
        labele = new JLabel [naziviLabela.length];

```



```

for (int i=0;i<labela.length;i++)
{
    labela[i]=new JLabel(naziviLabela[i]);
}
// Dodavanje polja za unos teksta
textPolja= new JTextField[4];
for (int i=0;i<textPolja.length;i++)
{
    textPolja[i]=new JTextField(10);
}
// Kreiranje dugmadi
dugme1=new JButton("Dodaj");
dugme1.setPreferredSize(new Dimension(114,20));
dugme2=new JButton("Učitaj spisak");
dugme2.setPreferredSize(new Dimension(114,20));
dugme3=new JButton("Sними spisak");
dugme3.setPreferredSize(new Dimension(114,20));
// Kreiranje check dugmeta
funkcijaDugmeta = new JCheckBox("Dodaj/Obriši",true);
// Kreiranje liste
spisak = new DefaultListModel ();
lista=new JList (spisak);
lista.setVisibleRowCount(3);
lista.setFixedCellWidth(120);
lista.setFixedCellHeight(39);
lista.setSelectionMode(ListSelectionModel.SINGLE_INTERVAL_SELECTION);
p2.add(labela[3]);
p2.add(textPolja[3]);
p2.add(labela[4]);
// Dodavanje liste
p2.add(lista);
// Dodavanje klizača na listu
p2.add(new JScrollPane(lista));
// Dodavanje tekstualnih polja na formu
for (int i=0;i<textPolja.length-1;i++)
{
    p1.add(labela[i]);
    p1.add(textPolja[i]);
}
// Dodavanje dugmadi na formu
p1.add(dugme1);
p1.add(funkcijaDugmeta);
p2.add(dugme2);
p1.add(dugme3);
// Dodavanje osluškivača akcije
Akcija klikNaDugme=new Akcija();
dugme1.addActionListener(klikNaDugme);
dugme2.addActionListener(klikNaDugme);
dugme3.addActionListener(klikNaDugme);
// Dodavanje osluškivača koji prati stanje check dugmeta
CheckDugme pratiFunkcijuDugmeta=new CheckDugme();
funkcijaDugmeta.addItemListener(pratiFunkcijuDugmeta);
// Dodavanje osluškivača koji prati unos karaktera
PratiPretragu prati=new PratiPretragu();
textPolja[3].addKeyListener(prati);
// Dodavanje osluškivača koji prati promjenu u listi
OsluskivacListe pratiListu=new OsluskivacListe();
lista.addMouseListener(pratiListu);
ljudi=new ArrayList <String>(10);
}
public void popuniListu(String ss)
{
    ss=ss.toLowerCase();
    spisak.clear();
    if (ss=="")
        for(String s:ljudi) spisak.addElement(s);
    else
        for(String s:ljudi) if(s.toLowerCase().indexOf(ss)>=0) spisak.addElement(s);
}

```

```

}

public void učitajSpisak()
{
    // Klasa koja omogućava izbor putanje do fajla
    JFileChooser fileopen = new JFileChooser();
    int ret = fileopen.showDialog(null, "Izaberi fajl");
    if (ret == JFileChooser.APPROVE_OPTION)
    {
        // Ako je fajl izabran, učitaj ga u listu
        String ime = fileopen.getSelectedFile().getName();
        String put = fileopen.getSelectedFile().getParent();
        fajl=put + "\\ " + ime;
        try
        {
            // Selektovani fajl
            File f=fileopen.getSelectedFile();
            Scanner sc=new Scanner(f);
            ljudi.clear();
            while (sc.hasNext())
            {
                String ss=sc.next()+"->" +sc.next()+"->" +sc.next();
                ljudi.add(ss);
            }
        }
        catch (Exception e)
        {
            System.out.println("Greška " + e);
        }
    }
}

// Implementacija oslušivača akcije
private class Akcija implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        // Ako je kliknuto na dugme za dodavanje/brisanje
        if (e.getSource()==dugme1)
        {
            String ss=textPolja[0].getText()+"->" +textPolja[1].getText()+"->" +textPolja[2].getText();
            if (funkcijaDugmeta.isSelected())
            {
                ljudi.add(ss);
            }
            else
            {
                for(int i=0;i<ljudi.size();i++) if(ljudi.get(i).equals(ss))
                {
                    ljudi.remove(i);
                    break;
                }
            }
            for (int i=0;i<textPolja.length;i++) textPolja[i].setText("");
            popuniListu("");
        }
        // Ako je kliknuto na dugme za učitavanje
        if (e.getSource()==dugme2)
        {
            učitajSpisak();
            popuniListu("");
        }
        // Ako je kliknuto na dugme za snimanje fajla
        if (e.getSource()==dugme3)
        {
            try
            {

```

```

        BufferedWriter b=new BufferedWriter (new FileWriter(fajl));
        for(int i=0;i<ljudi.size();i++)
        {
            String sss=ljudi.get(i);
            String pom[]=sss.split("->");
            b.write(pom[0]+" "+pom[1]+" "+pom[2]);
            b.newLine();
        }
        b.close();
    }
    catch(Exception ee)
    {
        System.out.println("Greška "+ ee);
    }
}
}

//Implementacija osluškivača koji prati unos karaktera
private class PratiPretragu extends KeyAdapter
{
    @Override
    public void keyTyped(KeyEvent e)
    {
        popuniListu(textPolja[3].getText());
    }
}

// Implementacija osluškivača koji prati stanje check dugmeta
private class CheckDugme implements ItemListener
{
    @Override
    public void itemStateChanged(ItemEvent e)
    {
        if (funkcijaDugmeta.isSelected()) dugme1.setText("Dodaj");
        else dugme1.setText("Obriši");
    }
}

//Implementacija osluškivača koji prati stanje liste
private class OsluskivacListe extends MouseAdapter
{
    @Override
    public void mouseClicked(MouseEvent ee)
    {
        if (ee.getSource()==lista)
        {
            String sss=spisak.getElementAt(lista.getSelectedIndex()).toString();
            String pom[]=sss.split("->");
            for (int i=0;i<pom.length;i++) textPolja[i].setText(pom[i]);
        }
    }
}

// Pokretanje programa
public static void main(String []args)
{
    Forma f=new Forma();
    f.setVisible(true);
}
}

```

Zadatak za samostalni rad: Modifikovati prethodni primjer tako što će se dodati obradu izuzetaka.